

- **What is CIMD?**

- CIMD is a dedicated data exchange protocol for connecting application programs to the Nokia Artuse service center, as well the short message service center (SMSC) and the USSD center (USSDC).

CIMD protocol packets are transmitted as text over a carrier protocol. The carrier protocol is usually TCP/IP; some operators may also still offer X.25 or plain serial modem connections.

- **Are there CIMD specifications for USSDC and SMSC connections?**

- The latest CIMD specification includes both SMSC and USSDC specific CIMD2 parameters in the same document.

- **What do I need for a CIMD connection? How do I get a CIMD connection?**

- Firstly, you need an ASE subscriber account at your operator's service center for your CIMD application. An application server entity (ASE) is the part within the Nokia SMS or USSD center to which the application connects using a CIMD protocol.

The application itself does not need to be aware of the ASE, but it needs a valid ASE subscriber account, username and password when sending messages to the service center. An ASE subscriber account is supplied by the service center operator that the application developer wants to use. This is a normal business contract between the developer and the operator. The ASE subscriber account is required to send messages, but the application may be developed on the application developer's site without a service center connection.

Please consult your operator partner's customer service for details on how to obtain CIMD access.

- **How do I use CIMD with a USSDC connection?**

- CIMD can be used for connecting applications to the Nokia Artuse USSD center. If the application involved does not use USSD specific functionalities, you can set your application to connect to USSDC instead of SMSC in the simplest case. See the latest CIMD specification for further details on USSDC-specific parameters.

Your application's ASE subscriber profile may be adjusted on the USSDC side so that USSDC-specific parameters are not passed to your application.

Note: USSDC is not a store-and-forward service center like SMSC. If the destination handset is not available to receive your USSD message, the submit fails and nothing is stored at USSDC.

See the Nokia Artuse USSD center product description for further details on USSDC functionality.

- **Can USSDC send empty messages to my application?**
- The USSDC center can send a submit-packet with an empty user data parameter value to an application.

033:<tab>

Example. Empty user data in a submit-packet from USSDC.

This can happen, for example, in the following scenario.

1. Handset sends '*20#' to USSDC's MSISDN 123.
2. USSDC runs with the "Strip Service Codes" parameter setup.
3. Since the "Strip Service Codes" parameter is on, service code 20 is clipped and the user data parameter value is empty when the application receives the CIMD packet.

Note: If your application tries to send a packet with no user data to USSDC, the packet will be discarded. Service centers are designed to silently drop messages which have no data. The above scenario might continue as follows.

4. The application that is connected to MSISDN 123 forwards the message received earlier to another MSISDN 456.
5. Since the user data is empty, USSDC drops the message because it cannot deliver empty messages.

Tip: Your application should be set so that it does not send CIMD packets with an empty user data field. It also should be enabled to receive CIMD packets with empty user data fields when they come through USSDC from any short message entity.

Important note for CIMD2 ADK version 1.0 and 1.0a users Please correct the file c2lASEPacket_c.cpp as follows:

line 231:

```
if(packet.GetToken(C2L_MATCH, token, 't')== (C2L_FOUND|C2L_TOKEN))
```

should be changed to:

```
if(packet.GetToken(C2L_MATCH, token, 't')&C2L_FOUND)
```

if you connect to USSDC and expect to receive packets with empty user data. Without the above fix, the CIMD2 ADK does not react to messages with empty user data.

- **How do I send 8-bit data that is more than 140 octets long to a handset?**
- You use the proper type of data header - i.e. the submit-packet's parameter 032. The Data is sent in consecutive submit-packets containing parameter 034's value. Parameter 030's value 245 indicates binary data.

```
<stx>03:021<tab>
021:123456789<tab>
032:0B0504158115810003000201<tab>
034:024A3995B1B1C..tone..data..81081581<tab>
030:245<tab><etx>
<stx>03:023<tab>
021:123456789<tab>
032:0B0504158115810003000202<tab>
034:08158158188108..tone..data..14000<tab>
030:245<tab><etx>
```

Example. Consecutive submit-packets for sending a ringing tone.

If the 8-bit data is not greater than 160 characters, the data can be sent in one message. In that case, the user data header is simpler.

```
032:06050415811581<tab>
```

Example. User data header for sending a ringing tone in one submit-packet.

Refer to the latest Smart Messaging specification for further details on the user data header that you will need.

Note: Your application must use an ASE subscriber account which has permission to use the appropriate user data header values for the binary data involved.

- **How do I connect to a service center with a dial-up modem?**
- Firstly, check what kind of dial-up modem connection your operator offers. If there is a TCP/IP protocol stack available through the dial-up line, you can use this to connect to the service center with the regular TCP/IP tools for modem line connections.

If you need to use a plain serial line modem connection, you need to check your own code for all error corrections as usual. Nokia Artuse service centers can offer plain modem connections for applications, but the application itself must maintain the data stream's integrity.

Tip: You can use the service provider's TCP/IP connection to connect to the Internet and then connect with TCP/IP to the service center.

Always consult with your operator to find the best connectivity solution for your application.

- **How do I connect to a service center with X.25?**
- An application can connect to the Nokia Artuse service center using an X.25 PAD connection. Consult your operator for details.
- **Is there a secure IP connection available for service center connections?**

- TCP/IP connections from the Internet to the Nokia Artuse service center can be secured using common firewall technologies. Consult your operator for available services.

Tip: You can find more details about Nokia Virtual Private Networks products for secure TCP/IP connections on the Nokia web site.

- **Why does the mobile telecommunication network appear not to accept the MSISDN that my application sends?**

Background to the question:

An application sends messages and the service center log shows that the mobile telecommunications network refuses to accept some of the messages.

- If your application uses an invalid MSISDN number, the service center accepts the message. The mobile network later replies to the SMSC with an error code and an attempt is made to deliver the message again according to the retry policy involved.

Tip: Check whether your application uses sub-numbering properly. Consult your operator about the service center's numbering plan if necessary.

- **How does sub-addressing work?**

- An application with MSISDN A (e.g. 1234) receives all messages that are sent to destination address AB (e.g. 12345678, 1234999 etc.).

If an application sends a message and sets the originator address to C (e.g. 999), the SMSC combines the application's MSISDN A (e.g. 1234) with the originator address issued by application C, making the originator address AC at SME. Thus, the handset receives AC (e.g. 1234999) as the application's originating address.

The sub-addressing feature described above concerns all ASE subscriber accounts without any special settings. Nevertheless, an operator can restrict message flows to certain MSISDN addresses according to the service center's numbering plan, which is implemented at the service center using number conversion rules. Consult your operator on the service center's numbering plan if necessary.

Note: When using sub-addressing, one must keep in mind that the maximum length of MSISDN is always 20.

Tip: A handset may attempt to send a message to your application using any destination address that has the application's MSISDN as its prefix. Your application should be aware of this when calculating the destination sub-address.

- **What are the possible error code values?**

- The error codes values for parameter 900 are integers in the range 0-999. Actual error codes can be found in the latest CIMD specification.

Tip: Your application should parse error codes that are one or more characters wide with non-digital integers at the end.

- **What does the start of the CIMD byte stream look like?**

Background to the question:

The CIMD client application seemed to halt immediately after opening the socket stream.

- At the beginning of the CIMD byte stream, there is normally a service center greeting message. The length of the greeting message may vary.

CIMD2-B ConnectionInfo: SessionId = 2 PortId = 0 Time = 991013134107 AccessType =
TCPIP_SOCKET PIN = 395688

Example. The CIMD session's greeting message.

The greeting message typically includes your CIMD session identifier (ID) and personal identifier number (PIN). If your operator supports CIMD trace, the ID and PIN are used when tracing the CIMD message flow.

Tip: Scan your socket stream for zero or other leading characters before the first CIMD packet.

- **Can two CIMD packets be delivered to an application without any intermediate characters?**

Background to the question:

Some CIMD client applications seemed not to receive a response packet from the service center. After a short period of time, the service center resent the response packet and the application continued its workflow.

- Two CIMD packets may immediately follow each other without any intermediate characters. Your application must assume that all CIMD packets start and end with protocol packet delimiters - normally STX> and <ETX>.

Zero or other non-protocol-packet-delimiter characters may lead or follow any CIMD protocol packet.

Two CIMD packets may immediately follow each other, e.g. when send-and-receive applications have been disconnected. When an application opens its communication channel again, then service center may send any pending status information packets or application terminating message packets, one immediately following another.

Tip: Parse your socket stream properly.

- **How do I force an SMS to be sent immediately?**

Refinement of the question:

The CIMD client application connects to the SMSC and it tries to send the SMS once. If the message cannot be delivered immediately, the SMS should be discarded.

- **What should the 050 parameter for the Submit-operation be in this case?**
- You can use
050:-1<tab>

This instructs the SMSC to try to deliver the message only once. Note that if another message is also pending for this subscriber, there is usually an attempt to deliver the first message and if this is successful, the new message is then delivered.

- **How can I check the status of a delivered SMS?**

Refinement of the question:

How can I implement an application that connects to the SMSC and sends a message status (04) enquiry for a given SMS, which was sent by another application?

- In CIMD version 2, there is no status report (SR) available unless the submitter requested one. Polling for status reports will always return the status code 0, which means that there is no status available. In order to get a status report it is necessary to add the 056 parameter to the submit messages, or you can use the profile settings. The latter method is easier as it does not require any application change.

The two-application setup would mean that the original sender application uses a send-only account, and the 'poll for results' application should be a send-and-retrieve application, where the subscriber addresses are similar.

For example

999 for the send-and-retrieve SR enquirer
9990 for the Information-In-Display send-only application

With this setup, the SRs will go to the send-and-retrieve application.

However, this polling is quite unnecessary as it causes a significant load on the system. The solution would be to activate delivery mode 2 for the send-and-retrieve application, which means that all SRs are immediately delivered to the application when they are available. There is actually no need for two separate applications, but it is possible to implement this functionality with two applications.

The increased system load is not significant in the latter case, as the system load increment is relatively low and the auto-delivery application is quite efficient.

Note that the SRs are not automatically delivered after login, unlike the messages.

- **What is the reason for error code 9?**

- Error code 9 implies that the SMSC has rejected the message. Messages can be rejected by the SMSC for the following reasons.

- The destination address is invalid, as in the example below.

021:hello<tab>

- The originating address and/or destination address is not valid.
 - A reserved or unsupported value is used in the DCS.
 - Too many messages have been sent to the same destination in a short period of time.

The event log can generally be used to find out why a message was rejected.

- **Should I wait for a response to my submit?**

- An application must not send a new submit before the previous submit-response has been received. If an application sends multiple messages without waiting for a response packet, it may not receive responses correctly.

Since the Nokia Artuse SMS center release SMSC 5B CD2, windowing is supported. With windowing you can initiate more than one operation before receiving responses. Default window size is 1.

- **What happens when a message is sent to a send-only CIMD application?**

Refinement of the question:

My send-only application is not designed to receive messages. What happens if someone tries to send an SMS to my application's service number?

- There is an attempt to deliver the message to the mobile terminal with the specified address. Normally this fails, as the number is probably not a valid MSISDN on the network.

Technically one can say that the SMSC handles the SMS in the usual manner when your application's ASE subscriber account is not designed to receive messages.

- **How many simultaneous CIMD logins are possible for one ASE subscriber account?**

- You can define how many simultaneous logins are possible by your ASE subscriber's profile file.

The default value is 10 simultaneous logins.

- **How can I trace my application's message flow when testing my CIMD application?**

- Check first from your operator that the anonymous tracing is supported!
The ASE supports remote tracing of connections by using a second TCP/IP connection from a telnet or web application, for example. Try the following:

```
telnet <ip-address> 9979
T <sesId> <pin>
```

Where <ip-address> is the same IP address that your application connects to, <sesId> and <pin> are the numbers contained in the first few bytes that are returned when you connect. The session ID and PIN are returned to you within the greeting message.

This will allow you to view details as your program executes its CIMD session. It is recommended that your CIMD program should have an option to sleep for 5-10 seconds or so after printing the greeting to a log file. You can then find the session ID and PIN in the log file and use a second session to connect to the operator port.

Other commands that are available through the operator port are, for example:

- C - close connection
- ? - print on-line help. All other commands - listed after choosing the '?' command - require a <login-id>, which is a different login ID to the CIMD client's login ID. In practice,
- T <sesId> <pin>
- C

are the only commands available.

Note that some telnet programs inaccurately translate <cr><nl> and mix up the output.

Consult your operator for further details on trace connection times and methods, if necessary.

- **Message delimiter characters seem to be lost somewhere?**
- Check with your operator whether you are using the correct port for your application's CIMD logins.

The CIMD protocol port is not necessarily the same every time. Each operator is free to choose the CIMD port that you are offered for your application. So the port number for CIMD application is not CIMD protocol specific but a specific issue for each service agreement.

See below for information on how you can type delimiter characters on the keyboard.

- **How can I type <STX> and <ETX> on my Unix terminal's keyboard when testing my CIMD application?**
- Try to issue the following Unix command before testing your software:

```
stty intr ^x
```


The ^C character (<CTRL>+C), which is the <ETX> end of message delimiter, is usually the Unix system's interrupt key. By redefining the Unix system's interrupt key to ^X (<CTRL>+X), you can type ^C at the end of a message.

The start of message delimiter <STX> can be typed on a standard Unix terminal using ^B (<CTRL>+B).

The parameter delimiter <tab> can normally be entered using the tabulator key.

Note that ordinary telnet applications require you to press <enter> before they resume with the input text line.

- **How can I encode Picture message user data (8-bit)**
- To get more information about smart messaging please study the Forum Nokia's Smart Messaging FAQs.

Please find example on sending a picture+text as a concatenate message in CIMD.

Header needs to be defined as header and the DCS is 245:

(032:xxxxxxx and 034:XXXX and 030:245)

032:0B0504158A00000003010301

034:300000045465737402010000481C01666666666666666666
9999999999999999800000000000000001400000006000E000024
000000E900310000280000031080CF3B80180000040041104440140
0000FFFE2F8B12024000000000538CAA0280000000006289C40180
00000000414140014000000000014280024000200000

030:245

032:0B0504158A00000003010302

034:014280028001F0000000A28001800FFE000000A500015
FFFFFFFFFEEA57FFA400AAA0000005500028201500440015D08A
1881024800040FF0201404100010003ABE00244000008200D555
88280101440001AAAAC0180000000003555560140010000806AA
AAB02400000000055555502800000000000000019999999999999

030:245

032:0B0504158A00000003010303

034:99996666666666666666666666666666

030:245

- **Why do I get an error message with error code 01 and the error text "Invalid character 'a' " when trying to use a special combination for the @ character?**
- Valid special combinations for GSM characters can be found in the following CIMD specification's chapter: Default user data character conversion.

The special combination for the @ character is _Oa (underscore, oh, a). If the message contains invalid characters after the underscore - and the combination is not a valid special combination - error 01 for invalid characters is raised.

- **What are the valid priority values?**
- Valid priority parameter values range from 1, the highest priority, to 9, the lowest priority. A priority parameter defines the relative importance of an SMS that is being sent from SMSC to the public land mobile network (PLMN). An attempt is made to deliver higher priority messages before delivering lower priority messages.

Your operator will give you the valid range of priorities to use. Follow your operator's recommendations if using priority parameters.

- **What is the CIMD Alive Operation packet used for?**
- By occasionally sending alive packets, your application informs the service center that your application is up and running. Please, do not load the IP network by sending spurious alive packets at frequent intervals.

Note: If the service center has a message to send to your application, the CIMD Deliver Message operation packet may arrive before the CIMD Positive Response to your alive packet. This is normal data communications behaviour. The CIMD Deliver Message operation packet may have been sent around the same moment that your application sent the alive packet.

The operator will inform you of the recommended time interval for sending alive packets. Follow your operator's recommendations for the sending intervals for alive packets.

- **What are the tariff classes used for?**
- Your operator will give you tariff class values that your application may use. They are used for separating the rates for different kinds of SMS or USSD dialogue messages that are sent from your application to the service center. Follow your operator's recommendations if using the tariff class parameter.
- **What does one do with characters outside the CIMD packet in the data stream?**
- Before receiving the first CIMD protocol packet from the service center, your application may receive a greeting string, if one is defined by your operator. Your application may save the greeting string in its own log file, for example.

Between two CIMD protocol packets from the service center, your application may receive an invalid character due, e.g., to problems on the carrier protocol line. Your application should always be able to omit invalid characters between the end of one CIMD protocol packet and the beginning of the next.

- **Why did I get some other CIMD packet when I waited for a CIMD response packet from the service center?**
- If the service center has a message to send to your application, the CIMD Deliver Message operation packet may arrive before the CIMD Positive Response to a previous CIMD packet sent by your application.

This is normal data communications behaviour. For example, the CIMD Deliver Message operation packet from the SMSC may have been sent around the same time as your application's previous CIMD Submit operation.

Note: This may happen with a CIMD Login operation packet. When the messaging center (MC) has accepted your application's login, the first pending message from the MC may be sent immediately and the CIMD Login Positive Response packet and CIMD Deliver Message operation packet may arrive in any order.

Remember also that no harm is done if your application does not immediately respond to the CIMD operation that originated at the MC. The MC will resend the CIMD operation packet in 60 seconds if the application does not send a corresponding response packet.

- **How I can use characters from the GSM alphabet extension table in my application?**
- The GSM 7-bit default alphabet extension table is defined in the "Digital cellular telecommunications system (Phase 2+); Alphabets and language-specific information (GSM 03.38)" specification.

In CIMD user data parameters, the _XX special combination character (named Reserved) is reserved as an escape to an extension of the GSM 7-bit default alphabet table. If the receiving mobile station does not understand the meaning of this escape mechanism, it should display it as a space character.

The following characters are defined in an extension character table. The CIMD Reserved escape character and the second CIMD character are written immediately after each other in the CIMD user data parameter, with no white space separating the entities.

Character name	Visual	CIMD Special Combination	Combination's names
Vertical bar		_XX_!!	Reserved, Inverted !

Caret	^	_XX_gl	Reserved, Greek Alphabet Lambda
Euro symbol	€	_XXe	Reserved, Small Letter e
Opening brace	{	_XX(Reserved, Left Parenthesi
Closing brace	}	_XX)	Reserved, Right Parenthesis
Page break	N/A	_XXASCII(10)	Reserved, Line Feed (<i>ascii 10</i>)
Opening bracket	[_XX<	Reserved, Less than sign
Closing bracket]	_XX>	Reserved, Greater Than Sign
Tilde	~	_XX=	Reserved, Equals Sign
Backslash	'	_XX/	Reserved, Slash

- **What happens to forthcoming status reports if the application logs out?**

- In the Nokia Artuse SMS center release SMSC 5A CD1 and earlier status reports were not stored for applications that were not logged in. This meant that when an application logged in again, it was not able to receive status reports that were generated while it was logged off.

Since the Nokia Artuse SMS center release SMSC 5B CD1, status reports are also stored for applications that are not logged in. This means that when an application logs in again, it may receive status reports that were generated while it was logged off.

Please consult with your operator about status report handling recommendations for the messaging center involved. Sometimes, operators are opposed to the use of status reports, since they load both the messaging center's database and TCP/IP LAN traffic. The potential additional load of unsolicited status reports may be significant in some environments, so obey your operator's SMS traffic policies.

- **Can I use a name string as an application's originating address?**

- Since the Nokia Artuse SMS center release SMSC 5B CD1, alphanumeric originator addresses are supported. The associated CIMD Submit operation parameter number is 027.

Note: A string is shown on the mobile station as the sender of the SMS. Nevertheless, the user cannot answer that SMS by using the alphanumeric originator address.

Note: Alphanumeric originator address will have to be enabled in the user profile file.

- **How do I send text that is more than 160 characters long to a handset?**
- If the data to be sent is just text, it is not necessary to encode it to binary form. Just encode the header to indicate concatenation and insert the text (fitting length: 140 oct - size of UDH = remaining octets. Then "remaining octets * 8 / 7" indicates how many 7-bit characters may still be added) to the user data parameter (033). Then send the rest of the data in the next message (again with the UDH, of course). Even the dcs does not need to be present, since the data is normal text.
- **Where I need port addressing scheme?**
- Port addressing scheme is needed in cases that the mobile (destination) terminal needs to handle the incoming message in a special way. It is used by the destination terminal to identify the content type. For example, if a ringing tone is sent, the phone needs to recognise it as a ringing tone and handle it accordingly, and not to display the characters on the screen. The same applies for example to picture messages. If no such intelligence is required from the destination (e.g. it is another application), then port addressing element is not needed.